Adapter A4 detects ports P11, P8, P9, P10 & P5,

Adapter A5 detects ports P3 & P12.

Though they do not have any ports in common, adapters A1 and A4 are in the same virtual SAN, since they both can see one or more ports in common with other adapters (e.g., adapter 3).

A general approach for handling any degree of complexity is to create collections of ports that belong together, and then work with each collection to ensure that all the ports that make up the collection are associated with the same SAN. SAN assignment for each collection is based on the following rules:

1) If any port in a collection is already known (e.g., by the SAN manager 20) to be on an actual SAN, then all ports in the collection are assumed to be on that SAN and not on any virtual SANs.

2) If none of the ports in the collection are known (e.g., by SAN manager 20) to be on an actual SAN, then the virtual SAN for the port with the highest port number is used for all ports in that collection.

3) If none of the ports in the collection are known to be on an actual or virtual SAN, then a new virtual SAN is created and used for ports in the collection.

4) If as a result of the above steps, a previously created virtual SAN no longer has any ports associated with it, that virtual SAN is discarded.

A methodology for implementing these rules is depicted in FIGURE 24. A first step 311 is to create collections of ports that are on actual SANs or that form potential virtual SANs based on

scan information in the discover engine 40 database. This is done by traversing the database from hosts to internal controllers, gathering all of the controller ports and then making calls via the operating system, to determine which endpoint ports are seen by these ports. The controller ports and the 'seen' ports are all added to this initial collection, referred to here as the fromPortPool.

Once fromPortPool has been populated, the SAN manager 20 creates two more collections called comparePorts and tempcollection. ComparePorts is seeded with a port from fromPortPool and then populated with any other ports in fromPortPool that see any ports in common with the seed port. Tempcollection is initialized with the seed port and any ports seen by the seed port. The ports from fromPortPool that see any ports in common with ports in comparePorts are added to tempcollection, and the ports seen by these ports are also added to tempcollection. Checks are made to ensure that none of the collections – i.e., comparePorts and tempCollection -- contain any duplicates - i.e., a port is not added to a collection if it is already in it.

Once the action described in the preceding two paragraphs has been taken, tempcollection consists of a collection of ports that may constitute a virtual SAN. The procedure described in these paragraphs is repeated by the SAN manager 20 over and over again using new comparePort and tempcollection collections until fromPortPool is empty. This results in a collection of tempcollection port collections. The next steps are to cleanup/establish the correct SAN-Port relationships for every port in each tempcollection as described below.

In a second step 313, for each collection of ports from the first step, the manager 20 determines if any port in the collection is already known to belong to an actual SAN. This can be determined by reference to the aforementioned manager databases, e.g., the discover engine database or, preferably, the topology database. If so, in step 315, the manager 20 deletes all virtual SAN references for every port in that collection and designates them all as being part of that same actual SAN.

If no port in the collection is already known to be assigned to an actual SAN (as determined in step 313), the manager in step 317 determines whether a virtual SAN is currently assigned to any ports in the collection. If not, in step 319, the manager creates a new virtual SAN, tempSan, as associates it with every port in the collection, e.g., by populating the topology database.

If a virtual SAN had been assigned to any ports in the collection (as determined in step 317), the manager in step 321 (i) removes the SanPortRelationships identifier for every port in the SAN that is not in the collection, (ii) in step 323, the SAN manager goes through each port in the collection and removes all SanPortRelationships except for those that reference tempSan, and (iii) in step 325, the SAN manager 20 creates a new SanPortRelationship from tempSan to each port in tempcollection that does not already have a relationship to it.

In step 327, the manager 20 removes all virtual SANs that no longer have any ports.